



S1.04 Etape 2

Justification du SLR de la BD Titanic

Relation PORT

Application de la règle R0 : Cette relation était une entité dans le SEA. L'identifiant est devenu une clé primaire, et les autres attributs sont devenus des propriétés.

Relation PASSAGER

Application de la règle R0 : Cette relation était une entité dans le SEA. L'identifiant est devenu une clé primaire, et les autres attributs sont devenus des propriétés. Ensuite, on rajoute deux propriétés supplémentaires :

- Application de la règle R1 : On ajoute la clé étrangère PClass, car la cardinalité de la patte entre l'association Registration et l'entité PASSENGER est 1,1.
Complément : L'entité CLASS comporte un seul attribut qui est l'identifiant, elle peut donc être supprimée. PClass n'est donc plus une clé étrangère.
- Application de la règle R1 : On ajoute la clé étrangère PortID, car la cardinalité de la patte entre l'association boarding et l'entité PASSENGER est 1,1.

Relation OCCUPATION

Application de la règle R3 : l'association occupation a ses deux pattes de cardinalité 0,n. Il convient donc de la transformer en nouvelle relation dont la clé primaire est le couple (CabinCode, PassengerID), ces deux attributs étant respectivement des clés étrangères qui font référence à l'entité CABIN, et l'entité PASSENGER.

Complément : L'entité CABIN comporte un seul attribut qui est l'identifiant, elle peut donc être supprimée. CabinCode n'est donc plus une clé étrangère dans la relation OCCUPATION.

Relation SERVICE

Application de la règle R2 : la patte "serves" de l'association service a pour cardinalité 0,1, et l'association n'a pas encore été traitée. Il convient donc d'en faire une nouvelle relation dont la clé primaire est PassengerId_Dom (patte "serves") et les autres propriétés sont :

- PassengerId_Emp, clé étrangère qui fait référence à la relation PASSENGER, mais en prenant cette fois-ci la patte "served by"
- Role, propriété de l'association qui reste donc inchangée.

Relation CATEGORY

Application de la règle R0 : Cette relation était une entité dans le SEA. L'identifiant est devenu une clé primaire, et les autres attributs sont devenus des propriétés.

Relation LIFEBOAT

Application de la règle R0 : Cette relation était une entité dans le SEA. L'identifiant est devenu une clé primaire, et les autres attributs sont devenus des propriétés. Ensuite, on rajoute deux propriétés supplémentaires :

- Application de la règle R1 : On ajoute la clé étrangère Launching_Time, car la cardinalité de la patte entre l'association launching et l'entité LIFEBOAT est 1,1.
Complément : L'entité OBSERVED_TIME comporte un seul attribut qui est l'identifiant, elle peut donc être supprimée. Launching_Time n'est donc plus une clé étrangère.
- Application de la règle R1 : On ajoute la clé étrangère LifeBoatCat, car la cardinalité de la patte entre l'association classification et l'entité LIFEBOAT est 1,1.

Relation RECOVERY

Application de la règle R2 : une des pattes de l'association service a pour cardinalité 0,1, et l'association n'a pas encore été traitée. Il convient donc d'en faire une nouvelle relation dont la clé primaire est aussi une clé étrangère (puisqu'elle référence la clé primaire de LIFEBOAT) : LifeBoatId. L'autre propriété provient de l'autre patte de recovery : Recovery_Time. OBSERVED_TIME ayant été supprimé par le complément, Recovery_Time devient une simple propriété, et plus une clé étrangère.

Relation RESCUE

Application de la règle R2 : une des pattes de l'association rescue a pour cardinalité 0,1, et l'association n'a pas encore été traitée. Il convient donc d'en faire une nouvelle relation dont la clé primaire référence celle de la relation PASSENGER : PassengerID. L'autre propriété provient de l'autre patte de rescue : LifeBoatId. C'est donc une clé étrangère.

Traduction des contraintes en SQL

PORT	
PortId	PortId char(1) primary key CONSTRAINT c_portid CHECK (PortId IN ('C','Q','S'))
PortName	PortName varchar NOT NULL
Country	Country varchar NOT NULL
PASSENGER	
PassengerId	PassengerId int primary key (PassengerId)
Name	Name varchar NOT NULL
Sex	Sex varchar NOT NULL
Age	Age int
Survived	Survived int CONSTRAINT c_survived CHECK (Survived IN (0,1))
PClass	PClass int NOT NULL CONSTRAINT c_pclass CHECK (Pclass IN (1,2,3))
PortId	PortId char(1) foreign key (PortId) references PORT(PortId)
OCCUPATION	
PassengerId	PassengerId int foreign key (PassengerId) references PASSENGER(PassengerId)
CabinCode	CabinCode varchar
(Concerne les deux propriétés)	primary key (PassengerId,CabinCode)
SERVICE	
PassengerId_Dom	PassengerId_Dom int primary key foreign key (PassengerId_Dom) references PASSENGER(PassengerId)
PassengerId_Emp	PassengerId_Emp int NOT NULL

	foreign key (PassengerId_Emp) references PASSENGER(PassengerId)
Role	Role varchar NOT NULL
CATEGORY	
LifeBoatCat	LifeBoatCat varchar primary key CONSTRAINT c_lifeboatcat CHECK (LifeBoatCat IN ('standard', 'secours', 'radeau'))
Structure	Structure varchar NOT NULL CONSTRAINT c_structure CHECK (Structure IN ('bois', 'bois et toile'))
Places	Places int NOT NULL
LIFEBOAT	
LifeBoatId	LifeBoatId varchar primary key
LifeBoatCat	LifeBoatCat varchar foreign key (LifeBoatCat) references CATEGORY(LifeBoatCat)
Side	Side varchar NOT NULL
Position	Position varchar NOT NULL CONSTRAINT c_position CHECK (Position IN ('babord', 'tribord'))
Location	Location varchar NOT NULL DEFAULT 'pont' CONSTRAINT c_location CHECK (Location IN ('avant', 'arriere'))
Launching_Time	Launching_Time Time NOT NULL
RECOVERY	
LifeBoatId	LifeBoatId varchar primary key foreign key (LifeBoatId) references LIFEBOAT(LifeBoatId)
Recovery_Time	Recovery_Time Time NOT NULL
RESCUE	
PassengerId	PassengerId int primary key foreign key (PassengerId) references PASSENGER(PassengerId)
LifeBoatId	LifeBoatId varchar NOT NULL